Certification Path Validation

References: X.509 sections: 8, 8.1, 12.4.1, 12.4.3, and 13.4
RFC 2459 sections: 1, 3.2, 4.1.2.4, 4.2.1.5, 4.2.1.10, 4.2.1.11, 4.2.1.12, 4.2.2.1, 6, 6.1, and 10
FPKI Profile sections: 1.1, 1.2, 1.2.6.2, 1.3, 1.4, App. A, and App. C
MISPC sections: 3.1.3.1, 3.1.3.3, 3.2.2, 3.2.3, and 3.3[1]
DII MA PKI Functional Specification sections: 3.3.3.1, and 3.3.3.4

Implementation under analysis:

Analysis Date:

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
Given a hierarchy of CAs, can each CA in the hierarchy store one certificate and one reverse certificate corresponding to its superior CA?		
Can CAs store user certificates and certification path information using the following ASN.1 data types? **Certificates ::= SEQUENCE {** **userCertificate Certificate,** **certificationPath ForwardCertificationPath OPTIONAL }** **CertificationPath ::= SEQUENCE {** **userCertificate Certificate,** **theCACertificates SEQUENCE OF CertificatePair OPTIONAL }** **ForwardCertificationPath ::= SEQUENCE OF CrossCertificates** **CrossCertificates ::= SET OF Certificate**		
Does the certification path development find a set of certificates that provide a chain of trust from the trusted CA to the end entity?		
Is this development done in the direction of trust, i.e., from the trusted CA to the end entity?		
Can a path be developed by starting at the end-entity and build a certificate chain back towards the user's trusted CA?		
Does the path development software attempt to find alternate paths, e.g., cross-certificates, if no certificate is returned to its certificate request?		

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
Does the path development software use the information provided by the path validation software to find alternate paths?		
Is the path development software able to process either the key identifier or the certificate issuer plus serial number form of authorityKeyIdentifier if this extension is used to find certification paths?		
Does the certificate user accept a self-signed certificate as a trusted certificate in the certification path?		
If two users have communicated before and have each other's certificates, do they authenticate without needing to validate the certification path?		
If two certificate users that want to authenticate are served by the same certification authority, can the certification path be established by the users obtaining each other's certificates?		
Given a hierarchy of CAs, can a certificate user store the public keys, certificates and reverse certificates of all certification authorities between itself and the root?		
If a certificate user frequently communicates with users certified by a particular other CA, can that user learn the certification path to that CA and the return certification path from that CA?		
If CAs have cross-certified one another, can the certificate user use this to establish the certification path?		
Having learned the certificate from the certification path, does the certificate user check the validity of the received certificate prior to authenticating the other user?		
When a key compromise or CA failure occurs for a trusted CA, is the user able to select other trusted CAs to provide to the path validation software?		
Does certification path validation processing occur in automated self-contained software free from user access or interference?		
Can the path validation software operate without local user participation?		
Does the path validation software distinguish EE certificates from CA certificates?		
Does the path validation software start with the CA that issued the local user's certificate?		
If the issuerUniqueIdentifier and subjectUniqueIdentifier fields are present in the base certificate, and non-empty, is the path validation software capable of parsing the unique identifiers and making comparisons, or is the certificate rejected?		2
Does the path validation software automatically apply the pathLenConstraint component of the basic constraint extension?		

REQUIREMENT FROM STANDARDS	**MET (Y/N/na)**	**NOTES**
Does the path validation software interpret the pathLenConstraint field as the maximum number of CA certificates that may follow this certificate in a certification path?		
When pathLenConstraint does not appear, does the path validation software interpret its absence as placing no limit to the allowed certification path length?		
Does the path validation software automatically check that the subjects of certificates are not located in an inappropriate name space as defined in the name constraint extension?		
Can the path validation software use the referenced CA Issuers description of the Authority Information Access extension to select a valid certification path?		
Can path validation software operate without reliance on a trusted local database of policy description information?		
Is the path validation software capable of establishing the path through multiple policy domains?		
Does the certificate user maintain an authenticated copy of the root-CA certificate approved for use by local security policy?		
Are the following inputs to the path validation software provided: a) a set of certificates comprising a certification path; b) a trusted (root-CA certificate) public key value or key identifier to verify the first certificate in the certification path; c) an *initial-policy-set*; d) an *initial-explicit-policy* indicator value (3) (4); e) an *initial-policy-mapping-inhibit* indicator value (3) (4); f) current date/time; and g) the time, T, for which the validity of the path should be determined.		3 4
Does the path validation software makes use of the following set of state variables: a) *user-constrained-policy-set*; b) *authority-constrained-policy-set*; c) *permitted-subtrees*; d) *excluded-subtrees*; e) *explicit-policy-indicator*; f) *policy-mapping-inhibit-indicator*; g) *explicit-policy-pending* constraint, h) *policy-mapping-inhibit-pending* constraint, i) *algorithm*(5), and j) *parameter*(5).		5

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
Does the path validation software perform the following: a) Initialize the *user-constrained-policy-set* variable to the value of *initial-policy-set*; b) Initialize the *authority-constrained-policy-set* variable to the value *any-policy*; c) Initialize the *permitted-subtrees* variable to *unbounded*; d) Initialize the *excluded-subtrees* variable to an empty set; e) Initialize the *explicit-policy-indicator* to the *initial-explicit-policy* value; f) Initialize the *policy-mapping-inhibit-indicator* to the *initial-policy-mapping-inhibit* value; h) Initialize the two *pending* constraint indicators to unset; and i) Initialize the algorithm and parameters state variables to the *subjectPublicKeyInfo* algorithm and parameters respectively of the user's Root-CA signature certificate.		6
If the certificate policies extension is critical, does the path validation software interpret this extension (including the optional qualifier), or reject the certificate?		
For each certificate in the path, does the path validation software check that the certificate was signed using the subject public key from the previous certificate?		
If validation fails do to inability to verify the digital signature on the end certificate, does the process terminate and return an indication that this was the reason for failure?		7
For each certificate in the path, does the path validation software check that the certificate validity period includes time T?		
If validation fails on this check, does the process terminate and return an indication that this was the reason for failure?		8
For each certificate in the path, does the path validation software check that the certificate is not revoked at time T or on a hold status that began prior to time T?		9
If validation fails on this check, does the process terminate and return an indication that this was the reason for failure?		
For each certificate in the path, does the path validation software check that the issuer and subject distinguished names chain correctly?		10
If validation fails on this check does the process terminate and return an indication that this was the reason for failure?		
For each certificate in the path, does the path validation software check that the subject name and subjectAltName extension (critical or noncritical) is consistent with the *permitted-subtrees* state variable?		

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
For each certificate in the path, does the path validation software check that the subject name and subjectAltName extension (critical or noncritical) is consistent with the *excluded-subtrees* state variable?		
If validation fails because the end certificate subject name is not within the name-space of *permitted-subtrees* and is within the name-space of *excluded-subtrees*, does the process terminate and return an indication that was the reason for failure?		
For each certificate in the path, does the path validation software check that, if the *explicit-policy-indicator* is set, a policy identifier in the certificate is in the initial policy set?		
For each certificate in the path, does the path validation software not map the policy identifier in the certificate when the *policy-mapping-inhibit-indicator* is set?		
For each certificate in the path, if the certificate policies extension is present and critical, does the path validation software verify a non-null value for *authority-constrained-policy-set*?		
For each certificate in the path, if the certificate policies extension is present and critical, does the path validation software compute the intersection of the policies in that extension and the *authority-constrained-policy-set*? Is the result put in as the new value of *authority-constrained-policy-set*?		
For each certificate in the path, does the path validation software verify a non-null intersection between *authority-constrained-policy-set* and *user-constrained-policy-set*?		
If validation fails do to an empty intersection of *authority-constrained-policy-set* and *user-constrained-policy-set* at the end certificate, does the process terminate and return an indication that was the reason for failure?		11
For the self-signed and intermediate certificates, does the path validation software recognize and process any other critical extension present?		
For the self-signed and intermediate certificates, does the path validation software verify it as a CA certificate?		
If validation fails do to an intermediate certificate with the BC extension absent, does the process terminate and return an indication that was the reason for failure?		
If validation fails do to an intermediate certificate with the cA component absent from the BC extension, does the process terminate and return an indication that was the reason for failure?		
If validation fails do to an intermediate certificate with the BC extension cA component set to false, does the process terminate and return an indication that was the reason for failure?		

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
If validation fails do to the certification path violating an intermediate certificate BC pathLenConstraint, does the process terminate and return an indication that was the reason for failure?		
For the self-signed and intermediate certificates with permittedSubtrees present, does the path validation software set the *permitted-subtrees* to the intersection of its previous value and the value indicated in the extension field?		
For the self-signed and intermediate certificates with excludedSubtrees present, does the path validation software set the *excluded-subtrees* to the union of its previous value and the value indicated in the extension field?		
For the self-signed and intermediate certificates with a critical key usage extension, does the path validation software check the keyCertSign bit is set?		
If validation fails because of this, does the software terminate the process and return an indication that this was the reason for failure?		
For each certificate in the path, does the path validation software check that the algorithm state variable is equal to the values in the *signature* field and in the *SIGNED* macro?		
If not all three are equal, the certificate is rejected and is another certificate path sought?		
If a certificate is encountered which has parameters in the *signature* field or the *SIGNED* macro, does the path validation software check that they are identical to the parameters state variable?		
If they are not identical, is the certificate is rejected and an alternate certificate path sought?		
If the algorithm state variable is different from the value of *algorithm* in the *subjectPublicKeyInfo* field of the certificate, is the algorithm state variable set to that value? Is the parameters state variable set to "null"?		
If the *subjectPublicKeyInfo* field of the certificate contains public key parameters, is the parameters state variable set to that value?		
Does path validation fail if no alternate certification path is found?		
Are the public key parameter values at the end of a successful chain validation the parameters used to verify the end entity signatures?		
Does the path validation software return an indication of validation success or failure?		
Can the certificate user not choose to use the certificate despite a certification path validation?		12
On validation success, does the path validation software terminate and return either the set of policies contained in *authority-constrained-policy-set*, within the constraints and with the qualifiers encounter in the certification path?		

REQUIREMENT FROM STANDARDS	**MET (Y/N/na)**	**NOTES**
If *any-policy* is not returned by the *authority-constrained-policy-set*, does the certificate user use the certificate according to one of the policies returned, applying all qualifiers for that policy?		
If *any-policy* is returned by the *authority-constrained-policy-set*, does the certificate user apply all qualifiers encountered in the certification path?		
On validation success, does the certificate user apply the policy mapping encountered in the certification path?		
On validation success, does the application software display all user notices in all certificates of the certification path used, excluding duplicates?		
If validation fails at the end certificate because no members of *user-constrained-policy-set* appear in the certificate policies field, does the process terminate and return an indication that was the reason for failure?		13

REQUIREMENT FROM STANDARDS	**MET (Y/N/na)**	**NOTES**
Upon process termination, are the following actions performed for each intermediate certificate: a) If the nameConstraints extension with a permittedSubtrees component is present in the certificate, set the *permitted-subtrees* state variable to the intersection of its previous value and the value indicated in the certificate extension. b) If the nameConstraints extension with an excludedSubtrees component is present in the certificate, set the *excluded-subtrees* state variable to the union of its previous value and the value indicated in the certificate extension. c) If *explicit-policy-indicator* is not set: ▪ If the *explicit-policy-pending* indicator is set, decrement the corresponding *skip-certificates* value and, if this value becomes zero, set *explicit-policy-indicator*. ▪ If the requireExplicitPolicy constraint is present in the certificate perform the following: For a SkipCerts value of 0, set *explicit-policy-indicator*. For any other requireExplicitPolicy SkipCerts value, set the *explicit-policy-pending* indicator (if previously unset), and set the corresponding *skip-certificates* value to the lesser of the requireExplicitPolicy SkipCerts value and the previous *skip-certificates* value (if the *explicit-policy-pending* indicator was already set)(14). d) If *policy-mapping-inhibit-indicator* is not set: ▪ process any policy mapping extension with respect to policies in the *user-constrained-policy-set* and add appropriate policy identifiers to the *user-constrained-policy-set* . ▪ process any policy mapping extension with respect to policies in the *authority-constrained-policy-set* and add appropriate policy identifiers to the *authority-constrained-policy-set* . ▪ if the *policy-mapping-inhibit-pending* indicator is set, decrement the corresponding *skip-certificates* value and, if this value becomes zero, set the *policy-mapping-inhibit-indicator*. ▪ If the inhibitPolicyMapping constraint is present in the certificate, perform the following: For a SkipCerts value of 0, set the *policy-mapping-inhibit-indicator*. For any other SkipCerts value, set the *policy-mapping-inhibit-pending* indicator (if previously unset), and set the corresponding *skip-certificates* value to the lesser of the SkipCerts value and the previous *skip-certificates* value (if the *policy-mapping-inhibit-pending* indicator was already set)(14).		14

REQUIREMENT FROM STANDARDS	MET (Y/N/na)	NOTES
Does the implementation support the following certificate constructs: **Certificates ::= SEQUENCE {** **userCertificate Certificate,** **certificationPath ForwardCertificationPath OPTIONAL}** **ForwardCertificationPath::= SEQUENCE OF CrossCertificates** **CrossCertificates ::= SET OF Certificate** **CertificationPath ::= SEQUENCE {** **userCertificate Certificate,** **theCACertificates SEQUENCE OF CertificatePair OPTIONAL}** **CertificatePair ::= SEQUENCE {** **forward [0] Certificate OPTIONAL,** **reverse [1] Certificate OPTIONAL** *-- at least one of the pair shall be present --* **}**		
Does the certificate user validate the certification path for an attribute certificate?		15

Does the implementation support the following attribute certificate path construct:
AttributeCertificationPath ::= SEQUENCE {
attributeCertificate AttributeCertificate,
acPath SEQUENCE OF ACPathData OPTIONAL }
ACPathData ::= SEQUENCE {
certificate [0] Certificate OPTIONAL,
attributeCertificate [1] AttributeCertificate OPTIONAL }		

Findings and supporting information:

Given that the DOD PKI system is being established at the v3 level, is the following (X.509, 12.4.1) applicable regarding the constraint extensions (BS, NC, and PC), i.e., is there going to be any interoperation with older 509 versions:

> Certificate extension fields need to be backward-compatible with the unconstrained certification path approach system as specified in earlier editions of ITU-T Rec. X.509 | ISO/IEC 9594-8.

RFC 2459 basic path validation procedure assumes one or more self-signed trusted CA certificates. In addition, that validation paths begin with most-trusted CAs (as defined by policy).

1. The certification path validation procedure specified in section 12.4.3 of X.509 is mandated by the FPKI MISPC. The X.509 calls only for functional equivalence. RFC 2459 also

bases its procedure on X.509, 12.4.3, and calls for only functional equivalence.

2. The requirement is as stated from section 1.4 of the FPKI. FPKI section 1.1 seems to contradict his by calling for the path validation software to ignore the issuerUniqueIdentifier and subjectUniqueIdentifier fields if they are present. RFC 2459 section 4.1.2.8 also calls for parsing these identifiers but not in the context of certificate path processing. RFC 2459 section 6.1 requires the processing of the identifiers if they are present.

3. RFC 2459 requires the initial value is the number of certificates in the certification path plus one.

4. FPKI specification sections 1.2.6.2 and 1.4 call for setting the initial-explicit-policy indicator to TRUE and the initial-policy-mapping-inhibit indicator to FALSE. This is in apparent conflict with the RFC 2459, which calls for integer values.

5. These state variables are called for in FPKI Appendix A for use in verifying DSA signatures.

6. RFC 2459 does not call for actions a) and g).

7. The last certificate in the path is considered the *end certificate*; the other certificates in the path are considered *intermediate certificates*.

8. FPKI section 1.4 permits the continued processing of an expired certificate as a vendor design decision. If processing is allowed to continue, for each expired certificate the application is to notify the user:

a) Which certificate is outside the interval;
b) When the certificate was, or will be valid;
c) Warned that the certificate may have been revoked and may not appear on a CRL;
d) That the signature may not be valid; and
e) Require the user to choose to proceed with the processing.

Real-time protocols (e.g. web page access) shall never be permitted to process expired certificates. The validity period of the self-signed certificate is not checked.

9. FPKI section 1.4 requires that revocation be checked for each certificate in the path, by verifying that each certificate serial number does not appear on the certificate issuer’s CRL or CRL indicated by the CRL Distribution Point extension.

10. Name chaining is performed by matching the issuer distinguished name in one certificate with the subject name in a CA certificate. That is, the issuer of a certificate was the subject of the previous certificate in the path. RFC 2459 section 4.2.1.7 implies that chaining applies also to the alternative name extensions.

11. Prior to determining this intersection, *authority-constrained-policy-set* value is updated.

12. Perhaps because of values of policy qualifiers or other information in the certificate.

13. Only when the *explicit-policy-indicator* is set and the certificate policies extension is present. An acceptable policy is a policy required by the user of the certification path or a policy that has been declared equivalent through policy mapping.

14. RFC 2459 text states (p56):

> If requireExplicitPolicy is present and has value r, the explicit policy state variable is set to the minimum of its current value and the sum of r and i (the current certificate in the sequence).
>
> If inhibitPolicyMapping is present and has value q, the policy mapping state variable is set to the minimum of its current value and the sum of q and i (the current certificate in the sequence)

The values "r" and "q" are SkipCerts from the policy constraint extension. The RFC gives "i" the value equal to the certificate's sequential place in the path, e.g., i=5 for the fifth certificate. This is at odds with X.509, which does not factor in the certificates location in the path. It chooses the lessor of SkipCerts or the previous pending constraints *skip-certificates* value. In addition, if SkipCerts is zero, X.509 calls for the setting of a policy indicator, i.e., an explicit policy is required or policy mapping is no longer permitted. The RFC prevents this from happening because SkipCerts (a.k.a. r or q) and i are summed together. As the path is traveled this sum would always be increasing, and it would not reflect the intent of the certificate's policy constraint. For example, the third certificate has a requireExplicitPolicy SkipCerts of five. If the previous *explicit-policy-pending skip-certificates* value was 10, the new *skip-certificates* value would be 8. This means three additional certificates may appear in the path before an explicit policy is required.

15. A subject may have multiple attribute certificates associated with each of its public key certificates. There is no requirement that the same CA create both the public key certificate and attribute certificate(s) for a user; in fact, separation of duties will frequently dictate otherwise. This requires multiple certificate path validations and the burden of this should be considered when defining local policy.

Recommendations for Standards Work: